

Attorney Docket No. 84513
Customer No. 23523

CHAIN RULE PROCESSOR

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN THAT PAUL M. BAGGENSTOSS, citizen of the United States of America, employee of the United States Government and resident Newport, County of Newport, State of Rhode Island, has invented certain new and useful improvements entitles as set forth above of which the following is a specification:

JAMES M. KASISCHKE, ESQ.
Reg. No. 36562
Naval Undersea Warfare Center
Division Newport
Newport, RI 02841-1708
TEL: 401-832-4736
FAX: 401-832-1231

I hereby certify that this correspondence is being deposited with the U.S.
Postal Service as U.S. EXPRESS MAIL, Mailing Label No. EV326644774US
In envelope addressed to: Commissioner for Patents, Alexandria, VA
22313 on 27 Feb 2004
(DATE OF DEPOSIT)

James M. Kasische
APPLICANT'S ATTORNEY

27 Feb 2004
DATE OF SIGNATURE

1 Attorney Docket No. 84513

2

3

CHAIN RULE PROCESSOR

4

5

STATEMENT OF GOVERNMENT INTEREST

6

7

8

9

The invention described herein may be manufactured and used by or for the Government of the United States of America for governmental purposes without the payment of any royalties thereon or therefor.

10

11

BACKGROUND OF THE INVENTION

12

(1) Field of the Invention

13

14

15

16

17

This invention generally relates to a signal classification system for classifying an incoming data stream. More particularly, the invention relates to a modularized classifier system that can be used for easily assembling different classifiers.

18

(2) Description of the Prior Art

19

20

21

22

23

24

25

26

27

In order to determine the nature of an incoming signal, the signal type must be determined. A classifier attempts to classify a signal into one of M signal classes based on features in the data. M-ary classifiers utilize neural networks for extracting these features from the data. In a training stage the neural networks incorporated in the classifier are trained with labeled data allowing the neural networks to learn the patterns associated with each of the M classes. In a testing stage, the classifier is tested against

1 unlabeled data based on the learned patterns. The performance
2 of the classifier is defined as the probability that a signal
3 is correctly classified.

4 The so-called M-ary classification problem is that of
5 assigning a multidimensional sample of data $x \in R^N$ to one of M
6 classes. The statistical hypothesis that class j is true is
7 denoted by H_j , $1 \leq j \leq M$. The statistical characterization of
8 x under each of the M hypotheses is described completely by
9 the probability density functions (PDFs), written

10 $p(x|H_j), 1 \leq j \leq M$. Classical theory as applied to the problem
11 results in the so-called Bayes classifier, which simplifies to
12 the Neyman-Pearson rule for equiprobable prior probabilities:

$$13 \quad j^* = \arg \max_j p(x|H_j). \quad (1)$$

14 Because this classifier attains the minimum probability of
15 error of all possible classifiers, it is the basis of most
16 classifier designs. Unfortunately, it does not provide simple
17 solutions to the dimensionality problem that arises when the
18 PDFs are unknown and must be estimated. The most common
19 solution is to reduce the dimension of the data by extraction
20 of a small number of information-bearing features $z = T(x)$,
21 then recasting the classification problem in terms of z :

$$22 \quad j^* = \arg \max_j p(z|H_j). \quad (2)$$

23 This leads to a fundamental tradeoff: whether to discard
24 features in an attempt to reduce the dimension to something

1 manageable or to include them and suffer the problems
2 associated with estimating a PDF at high dimension.
3 Unfortunately, there may be no acceptable compromise.
4 Virtually all methods which attempt to find decision
5 boundaries on a high-dimensional space are subject to this
6 tradeoff or "curse" of dimensionality. For this reason, many
7 researchers have explored the possibility of using class-
8 specific features.

9 The basic idea in using class-specific features is to
10 extract M class-specific feature sets $z_j = T_j(x)$, $1 \leq j \leq M$ where
11 the dimension of each feature set is small, and then to arrive
12 at a decision rule based only upon functions of the lower
13 dimensional features. Unfortunately, the classifier modeled on
14 the Neyman-Pearson rule

$$15 \quad j^* = \arg \max_j p(z_j | H_j). \quad (3)$$

16 is invalid because comparisons of densities on different
17 feature spaces are meaningless. One of the first approaches
18 that comes to mind is to compute for each class a likelihood
19 ratio against a common hypothesis composed of "all other
20 classes." While this seems beneficial on the surface, there is
21 no theoretical dimensionality reduction since for each
22 likelihood ratio to be a sufficient statistic, "all features"
23 must be included when testing each class against a hypothesis
24 that includes "all other classes." A number of other
25 approaches have emerged in recent years to arrive at
26 meaningful decision rules. Each method makes a strong

1 assumption (such as that the classes fall into linear
2 subspaces) that limits the applicability of the method or else
3 uses *ad hoc* method of combining the likelihoods of the various
4 feature sets.

5 Prior art methods include the following. A method used
6 in speech recognition (Frimpong-Ansah, K. Pearce, D. Holmes,
7 and W. Dixon, "A stochastic/feature based recognizer and its
8 training algorithm," in *Proc. ICASSP*, vol. 1, 1989, pp. 401-
9 404.) uses phoneme-specific features. While, at first, this
10 method appears to use class-specific features, it is actually
11 using the same features extracted from the raw data but
12 applying different models to the time evolution of these
13 features.

14 A method of image recognition (E. Sali and S. Ullman,
15 "Combining class-specific fragments for object
16 classification," in *Proc. British Machine Vision Conf.*, 1999,
17 pp. 203-213.) uses class-specific features to detect various
18 image "fragments." The method uses a nonprobabilistic means of
19 combining fragments to form an image.

20 A method has been proposed that tests all pairs of
21 classes (S. Kumar, J. Ghosh, and M. Crawford, "A versatile
22 framework for labeling imagery with large number of classes,"
23 in *Proc. Int. Joint Conf. Neural Networks*, Washington, DC,
24 1999, pp. 2829-2833.). To be exhaustive, this method has a
25 complexity of $O(M^2)$ different tests and may be prohibitive for
26 large M . A hierarchical approach has been proposed based on a
27 binary tree of tests ("A hierarchical multiclassifier system

1 for hyperspectral data analysis," in *Multiple Classifier*
2 *Systems*, J. Kittler and F. Roli, Eds. New York: Springer,
3 2000, pp. 270-279). Implementation of the binary tree requires
4 initial classification into meta-classes, which is an approach
5 that is suboptimal because it makes hard decisions based on
6 limited information.

7 Methods based on linear subspaces (H. Watanabe, T.
8 Yamaguchi, and S. Katagiri, "Discriminative metric design for
9 robust pattern recognition," *IEEE Trans. Signal Processing*,
10 vol. 45, pp. 2655-2661, Nov. 1997. P. Belhumeur, J. Hespanha,
11 and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition
12 using class specific linear projection," *IEEE Trans. Pattern*
13 *Anal. Machine Intell.*, vol. 19, pp. 711-720, July 1997.) are
14 popular because they use the powerful tool of linear subspace
15 analysis. These methods can perform well in certain
16 applications but are severely limited to problems where when
17 the classes are separable by linear processing.

18 Support vectors (D. Sebal, "Support vector machines and
19 the multiple hypothesis test problem," *IEEE Trans. Signal*
20 *Processing*, vol. 49, pp. 2865-2872, Nov. 2001.) are a
21 relatively new approach that is based on finding a linear
22 decision function between every pair of classes.

23 The inventor has also developed a prior class specific
24 classifier, U.S. Patent No. 6,535,641, showing a class
25 specific classifier for classifying data received from a data
26 source. The classifier has a feature transformation section
27 associated with each class of data which receives the data and

1 provides a feature set for the associated data class. Each
2 feature transformation section is joined to a pattern matching
3 processor which receives the associated data class feature
4 set. The pattern matching processors calculate likelihood
5 functions for the associated data class. One normalization
6 processor is joined in parallel with each pattern matching
7 processor for calculating an inverse likelihood function from
8 the data, the associated class feature set and a common data
9 class set. The common data class set can be either calculated
10 in a common data class calculator or incorporated in the
11 normalization calculation. The inverse likelihood function is
12 then multiplied with the likelihood function for each
13 associated data class. A comparator provides a signal
14 indicating the appropriate class for the input data based upon
15 the highest multiplied result.

16 As evidenced by the various approaches, there is a strong
17 motivation for using class-specific features. Unfortunately,
18 classical theory as it stands requires operating in a common
19 feature space and fails to provide any guidance for a suitable
20 class-specific architecture.

21

22 SUMMARY OF THE INVENTION

23 Therefore, it is one purpose of this invention to provide
24 a class specific classifier.

25 Another purpose of this invention is a classifier
26 architecture having reusable modules.

1 Accordingly, there is provided a modularized classifier
2 which includes a plurality of class specific modules. Each
3 module has a feature calculation section, and a correction
4 section. The modules can be arranged in chains of modules
5 where each chain is associated with a class. The first module
6 in the chain receives raw input data and subsequent modules
7 act on the features provided by the previous module. The
8 correction section acts on the previously computed correction.
9 Each chain is terminated by a probability density function
10 evaluation module. The output of the evaluation module is
11 combined with the correction value of the last module in the
12 chain. This combined output is provided to a compare module
13 that indicates the class of the raw input data. The invention
14 may be implemented either as a device or a method operating on
15 a computer.

16

17 BRIEF DESCRIPTION OF THE DRAWINGS

18 The appended claims particularly point out and distinctly
19 claim the subject matter of this invention. The various
20 objects, advantages and novel features of this invention will
21 be more fully apparent from a reading of the following
22 detailed description in conjunction with the accompanying
23 drawings in which like reference numerals refer to like parts,
24 and in which:

25 FIG. 1 is a diagram illustrating the chain rule used in
26 this invention;

1 FIG. 2 is a block diagram of a first example of a
2 classifier implemented utilizing the preferred architecture of
3 the current invention;

4 FIG. 3 is a block diagram of a second example of a
5 classifier implemented utilizing an alternative architecture
6 of the current invention; and

7 FIG. 4 is a block diagram of an embodiment of a
8 classifier implemented utilizing another alternative
9 architecture of the current invention.

10

11 DESCRIPTION OF THE PREFERRED EMBODIMENT

12 It is well known how to write the PDF of x from the PDF
13 of z when the transformation is 1:1. This is the change of
14 variables theorem from basic probability. Let $z=T(x)$, where
15 $T(x)$ is an invertible and differentiable multidimensional
16 transformation. Then,

$$17 \quad p_x(x) = |J(x)|p_z(T(x)), \quad (4)$$

18 where $|J(x)|$ is the determinant of the Jacobian matrix of
19 the transformation

$$20 \quad J_{ij} = \frac{\partial z_i}{\partial x_j}. \quad (5)$$

21 What we seek is a generalization of (4) which is valid
22 for many-to-1 transformations. Define

23

$$24 \quad P(T, p_z) = \{p_x(x) : z = T(x) \text{ and } z \sim p_z(z)\}, \quad (6)$$

1 that is, $P(T, p_z)$ is the set of PDFs $p_x(x)$ which through $T(x)$
 2 generates PDF $p_z(z)$ on z . If $T(\cdot)$ is many-to-one, $P(T, p_z)$ will
 3 contain more than one member. Therefore, it is impossible to
 4 uniquely determine $p_x(x)$ from $T(\cdot)$ and $p_z(z)$. We can, however,
 5 find a particular solution if we constrain $p_x(x)$ such that for
 6 every transform pair (x, z) , we have:

$$7 \quad \frac{p_x(x)}{p_x(x|H_0)} = \frac{p_z(z)}{p_z(z|H_0)}, \quad (7)$$

8 or that the likelihood ratio (with respect to H_0) is the same
 9 in both the raw data and feature domains for some pre-
 10 determined reference hypothesis H_0 . We will soon show that
 11 this constraint produces desirable properties. The particular
 12 form of $p_x(x)$ is uniquely defined by the constraint itself,
 13 namely

$$14 \quad p_x(x) = \frac{p_z(x|H_0)}{p_z(z|H_0)} p_z(z); \text{ at } z = T(x). \quad (8)$$

15 The PDF projection theorem proves that (8) is, indeed, a
 16 PDF and a member of $P(T, p_z)$. Under this theorem let H_0 be some
 17 fixed reference hypothesis with known PDF $p_x(x|H_0)$. Let χ be
 18 the region of support of $p_x(x|H_0)$. In other words χ is the set
 19 of all points x where $p_x(x|H_0) > 0$. Let $z = T(x)$ be a continuous
 20 many-to-one transformation (the continuity requirement may be
 21 overly restrictive). Let Z be the image of χ under the
 22 transformation $T(x)$. Let $p_z(z|H_0)$ be the PDF of z when x is

1 drawn from $p_x(x|H_0)$. It follows that $p_z(z|H_0) > 0$ for all $z \in Z$.

2 Now, let be a any other PDF with the same region of support Z .

3 Then the function (8) is a PDF on χ , thus

4

$$5 \quad \int_{x \in \chi} p_x(x) dx = 1. \quad (9)$$

6

7 Furthermore, $p_x(x)$ is a member of $P(T, p_z)$.

8 The theorem shows that, provided we know the PDF under

9 some reference hypothesis H_0 at both the input and output of

10 transformation $T(x)$, if we are given an arbitrary PDF $p_z(z)$

11 defined on z , we can immediately find a PDF $p_x(x)$ defined on x

12 that generates $p_z(z)$. Although it is interesting that $p_x(x)$

13 generates $p_z(z)$, there are an infinite number of them, and it

14 is not yet clear that $p_x(x)$ is the best choice. However,

15 suppose we would like to use $p_x(x)$ as an approximation to the

16 PDF $p_x(x|H_1)$. Let this approximation be

17

$$18 \quad \hat{p}_x(x|H_1) \equiv \frac{p_x(x|H_0)}{p_z(z|H_0)} \hat{p}_z(z|H_1) \text{ at } z = T(x). \quad (10)$$

19

20 From the PDF projection theorem, we see that (10) is a PDF.

21 Furthermore, if $T(x)$ is a sufficient statistic for H_1 vs H_0 ,

22 then as $\hat{p}_z(z|H_1) \rightarrow p_z(z|H_1)$, we have

$$23 \quad \hat{p}_x(x|H_1) \rightarrow p_x(x|H_1). \quad (11)$$

1 This is immediately seen from the well-known property of the
 2 likelihood ratio, which states that if $T(x)$ is sufficient for
 3 H_1 versus H_0 :

$$4 \quad \frac{p_x(x|H_1)}{p_x(x|H_0)} = \frac{p_z(z|H_1)}{p_z(z|H_0)} \quad (12)$$

5 Note that for a given H_1 , the choice of $T(x)$ and H_0 are coupled
 6 so that they must be chosen *jointly*. In addition, note that
 7 the sufficiency condition is required for optimality, but is
 8 not necessary for (10) to be a valid PDF. Here, we can see the
 9 importance of the theorem. The theorem, in effect, provides a
 10 means of creating PDF approximations on the high-dimensional
 11 input data space without dimensionality penalty using low-
 12 dimensional feature PDFs and provides a way to optimize the
 13 approximation by controlling both the reference hypothesis H_0
 14 as well as the features themselves. This is the remarkable
 15 property of the theorem: that the resulting function remains a
 16 PDF whether or not the features are sufficient statistics.
 17 Since sufficiency means optimality of the classifier,
 18 approximate sufficiency means PDF approximation and
 19 approximate optimality.

20 The PDF projection theorem allows maximum likelihood (ML)
 21 methods to be used in the raw data space to optimize the
 22 accuracy of the approximation over T and H_0 as well as θ . Let
 23 $\hat{p}_z(z|H_1)$ be parameterized by the parameter θ . Then, the
 24 maximization

$$\max_{\theta, T, H_0} \left\{ \frac{p_x(x|H_0)}{p_z(z|H_0)} \hat{p}_z(z|H_1; \theta), z = T(x) \right\} \quad (13)$$

is a valid ML approach and can be used for model selection (with appropriate data cross-validation).

We now mention a useful property of (7). Let H_z be a region of sufficiency (ROS) of z , which is defined as a set of all hypotheses such that for every pair of hypotheses $H_{0a}, H_{0b} \in H_z$, we have

$$\frac{p_x(x|H_{0a})}{p_x(x|H_{0b})} = \frac{p_z(z|H_{0a})}{p_z(z|H_{0b})} \quad (14)$$

An ROS may be thought of as a family of PDFs traced out by the parameters of a PDF, where z is a sufficient statistic for the parameters. The ROS may or may not be unique. For example, the ROS for a sample mean statistic could be a family of Gaussian PDFs with variance 1 traced out by the mean parameter. Another ROS would be produced by a different variance. The "j-function"

$$J(x, T, H_0) \equiv \frac{p_x(x|H_0)}{p_z(T(x)|H_0)} = \frac{p(x|H_0)}{p(z|H_0)} \quad (15)$$

is independent of H_0 as long as H_0 remains within ROS H_z . Defining the ROS should in no way be interpreted as a sufficiency requirement for z . All statistics z have an ROS that may or may not include H_1 (it does only in the ideal case). Defining H_z is used only in determining the allowable range of reference hypotheses when using a data-dependent reference hypothesis. For example, let z be the sample

1 variance of x . Let $H_0(\sigma^2)$ be the hypothesis that x is a set of
 2 N independent identically distributed zero-mean Gaussian
 3 samples with variance σ^2 . Clearly, an ROS for z is the set of
 4 all PDFs traced out by σ^2 . We have

$$5 \quad p(x|H_0(\sigma^2)) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2} \sum_{n=1}^N x_n^2\right\} \quad (16)$$

6 and, since z is a $\chi^2(N)$ random variable (scaled by $1/N$)

$$7 \quad p(z|H_0(\sigma^2)) = \frac{N}{\sigma^2 \Gamma\left(\frac{N}{2}\right)} 2^{-N/2} \left(\frac{N_z}{\sigma^2}\right)^{N/2-1} \exp\left(-\frac{zN}{2\sigma^2}\right). \quad (17)$$

8 It is easily verified that the contribution of σ^2 is canceled
 9 in the J-function ratio.

10 Because $J(x, T, H_0(\sigma^2))$ is independent of σ^2 , it is possible
 11 to make σ^2 a function of the data itself, changing it with
 12 each input sample. In the example above, since z is the
 13 sample variance, we could let the assumed variance under H_0
 14 depend on z according to $\sigma^2 = z$.

15 However, if $J(x, T, H_0(\sigma^2))$ is independent of σ^2 , one may
 16 question what purpose does it serve to vary σ^2 . The reason is
 17 purely numerical. Note that in general, we do not have an
 18 analytic form for the J-function but instead have separate
 19 numerator and denominator terms. Often, computing $J(x, T, H_0(\sigma^2))$
 20 can pose some tricky numerical problems, particularly if x and
 21 z are in the tails of the respective PDFs. Therefore, our
 22 approach is to position H_0 to maximize the numerator PDF

1 (which simultaneously maximizes the denominator). Another
 2 reason to do this is to allow PDF approximations to be used in
 3 the denominator that are not valid in the tails, such as the
 4 central limit theorem (CLT).

5 In our example, the maximum of the numerator clearly
 6 happens at $\sigma^2 = z$ because z is the maximum likelihood
 7 estimator of σ^2 . We will explore the relationship of this
 8 method to asymptotic ML theory in a later section. To reflect
 9 the possible dependence of H_0 on z , we adopt the notation
 10 $H_0(z)$. Thus

$$11 \quad \hat{p}_x(x|H_1) \equiv \frac{p_x(x|H_0(z))}{p_z(z|H_0(z))} \hat{p}_z(z|H_1), \text{ where } z = T(x). \quad (18)$$

12 The existence of z on the right side of the conditioning
 13 operator $|$ is admittedly a very bad use of notation but is done
 14 for simplicity. The meaning of z can be understood using the
 15 following imaginary situation. Imagine that we are handed a
 16 data sample x , and we evaluate (10) for a particular
 17 hypothesis $H_0 \in \mathbf{H}_z$. Out of curiosity, we try it again for a
 18 different hypothesis of $H'_0 \in \mathbf{H}_z$. We find that no matter which
 19 $H_0 \in \mathbf{H}_z$ we use, the result is the same. We notice, however,
 20 that for an H_0 that produces larger values of $p_x(x|H_0(z))$ and
 21 $p_z(z|H_0(z))$, the requirement for numerical accuracy is less
 22 stringent. It may require fewer terms in a polynomial
 23 expansion or else fewer bits of numerical accuracy. Now, we
 24 are handed a new sample of x , but this time, having learned

1 our lesson, we immediately choose the $H_0 \in \mathcal{H}_z$ that maximizes
2 $p_x(x|H_0(z))$. If we do this every time, we realize that H_0 is now
3 a function of z . The dependence, however, carries no
4 statistical meaning and only has a numerical interpretation.
5 This is addressed below in the text differentiating a fixed
6 reference hypothesis from a variable reference hypothesis.

7 In many problems \mathcal{H}_z is not easily found, and we must be
8 satisfied with approximate sufficiency. In this case, there
9 is a weak dependence of $J(x, T, H_0)$ upon H_0 . This dependence is
10 generally unpredictable unless, as we have suggested, $H_0(z)$ is
11 always chosen to maximize the numerator PDF. Then, the
12 behavior of $J(x, T, H_0)$ is somewhat predictable. Because the
13 numerator is always maximized, the result is a positive bias.
14 This positive bias is most notable when there is a good match
15 to the data, which is a desirable feature.

16 We have stated that when we use a data-dependent or
17 variable reference hypothesis, we prefer to choose the
18 reference hypothesis such that the numerator of the J-function
19 is a maximum. Since we often have parametric forms for the
20 PDFs, this amounts to finding the ML estimates of the
21 parameters. If there are a small number of features, all of
22 the features are ML estimators for parameters of the PDF, and
23 there is sufficient data to guarantee that the ML estimators
24 fall in the asymptotic (large data) region, then the variable
25 hypothesis approach is equivalent to an existing approach.

1 based on classical asymptotic ML theory. We will derive the
2 well-known asymptotic result using (18).

3 Two well-known results from asymptotic theory are the
4 following. First, subject to certain regularity conditions
5 (large amount of data, a PDF that depends on a finite number
6 of parameters and is differentiable, etc.), the PDF $p_x(x; \theta^*)$
7 may be approximated by

$$9 \quad p_x(x; \theta^*) \cong p_x(x; \hat{\theta}) \exp \left\{ -\frac{1}{2} (\theta^* - \hat{\theta})' I(\hat{\theta}) (\theta^* - \hat{\theta}) \right\} \quad (19)$$

10

11 Where θ^* is an arbitrary value of the parameter $\hat{\theta}$ is the
12 maximum likelihood estimate (MLE) of θ , and $I(\theta)$ is the
13 *Fisher's information matrix* (FIM). The components of the FIM
14 for PDF parameters θ_k, θ_t are given by

$$15 \quad I_{\theta_k, \theta_t}(\theta) = -E \left(\frac{\partial^2 \ln p_x(x; \theta)}{\partial \theta_k \partial \theta_t} \right). \quad (20)$$

16

17 The approximation is valid only for θ^* in the vicinity of the
18 MLE (and the true value). Second, the MLE $\hat{\theta}$ is approximately
19 Gaussian with mean equal to the true value θ and covariance
20 equal to $I^{-1}(\theta)$ or

$$p_{\theta}(\hat{\theta}; \theta) \cong (2\pi)^{-P/2} \left| I(\hat{\theta}) \right|^{1/2} \exp \left\{ -\frac{1}{2} (\theta - \hat{\theta})' I(\hat{\theta}) (\theta - \hat{\theta}) \right\} \quad (21)$$

2

3 where P is the dimension of θ . Note that we use $\hat{\theta}$ in
 4 evaluating the FIM in place of θ , which is unknown. This is
 5 allowed because $I^{-1}(\theta)$ has a weak dependence on θ . The
 6 approximation is valid only for θ in the vicinity of the MLE.

7 To apply (18), $\hat{\theta}$ takes the place of z , and $H_0(z)$ is the
 8 hypothesis that $\hat{\theta}$ is the true value of θ . We substitute (19)
 9 for $p_x(x|H_0(z))$ and (21) $p_z(z|H_0(z))$. Under the stated conditions,
 10 the exponential terms in approximations (19), and (21) become
 11 1. Using these approximations, we arrived at

12

$$\hat{p}_x(x|H_1) = \frac{p_x(x; \hat{\theta})}{(2\pi)^{-P/2} \left| I(\hat{\theta}) \right|^{1/2}} \hat{p}_{\theta}(\hat{\theta}|H_1) \quad (22)$$

14

15 which agrees with the PDF approximation from asymptotic
 16 theory.

17 To compare (18) and (22), we note that for both, there is
 18 an implied sufficiency requirement for z and $\hat{\theta}$, respectively.
 19 Specifically, $H_0(z)$ must remain in the ROS of z , whereas $\hat{\theta}$ must
 20 be asymptotically sufficient for θ . However, (18) is more

1 general since (22) is valid only when all of the features are
 2 ML estimators and only holds asymptotically for large data
 3 records with the implication that $\hat{\theta}$ tends to Gaussian, whereas
 4 (18) has no such implication. This is particularly important
 5 in upstream processing, where there has not been significant
 6 data reduction, and asymptotic results do not apply. Using
 7 (18), we can make simple adjustments to the reference
 8 hypothesis to match the data better and avoid the PDF tails
 9 (such as controlling variance), where we are certain that we
 10 remain in the ROS of z . As an aside, we note that (10) with a
 11 fixed reference hypothesis is even more general since there is
 12 no implied sufficiency requirement for z .

13 In many cases, it is difficult to derive the J-function
 14 for an entire processing chain. On the other hand, it may be
 15 quite easy to do it for one stage of processing at a time. In
 16 this case, the chain rule can be used to good advantage. The
 17 chain rule is just the recursive application of the PDF
 18 projection theorem. For example, consider a processing chain

$$19 \quad \begin{matrix} T_1(x) & T_2(y) & T_3(w) \\ x & \rightarrow y & \rightarrow w & \rightarrow z \end{matrix} \quad (23)$$

20 The recursive use of (10) gives

$$21 \quad p_x(x|H_1) = \frac{p_x(x|H_0(y))}{p_y(y|H_0(y))} \frac{p_y(y|H'_0(w))}{p_w(w|H'_0(w))} \frac{p_w(w|H''_0(z))}{p_z(z|H''_0(z))} p_z(z|H_1) \quad (24)$$

22 where $y = T_1(x)$, $w = T_2(y)$, $z = T_3(w)$, and $H_0(y)$, $H'_0(w)$, $H''_0(z)$ are
 23 reference hypotheses (possibly data-dependent) suited to each
 24 stage in the processing chain. By defining the J-function of

1 each stage, we may write the above as

$$2 \quad p_x(x|H_1) = J(x, T_1, H_0(y)) J(y, T_2, H'_0(w)) \quad (25)$$

$$J(w, T_3, H''_0(z)) p_z(z|H_1).$$

3 There is a special embedded relationship between the
 4 hypotheses. Let H_y , H_w , and H_z be the ROSSs of y , w , and z ,
 5 respectively. Then, we have $H_z \subset H_w \subset H_y$. If we use variable
 6 reference hypotheses, we also must have
 7 $H''_0(z) \in H_z$, $H'_0(w) \in H_w$, and $H_0(y) \in H_y$. This embedding of the
 8 hypotheses is illustrated in FIG. 1. The condition $H_1 \in H_z$ is
 9 the ideal situation and is not necessary to produce a valid
 10 PDF. The factorization (24), together with the embedding of
 11 the hypotheses, we call the chain-rule processor (CRP).

12 We now summarize the various methods we have discussed
 13 for computing the J-function. For modules using a fixed
 14 reference hypothesis, care must be taken in calculation of the
 15 J-function because the data is more often than not in the
 16 tails of the PDF. For fixed reference hypotheses, the J-
 17 function is

$$18 \quad J(x, T, H_0) = \frac{p_x(x|H_0)}{p_z(z|H_0)}. \quad (26)$$

19 The numerator density is usually of a simple form, so it is
 20 known exactly. The denominator density $p_z(z|H_0)$ must be known
 21 exactly or approximated carefully so that it is accurate even
 22 in the far tails of the PDF. The saddlepoint approximation
 23 (SPA) provides a solution for cases when the exact PDF cannot

1 be derived but the exact moment-generating function is known.

2 The SPA is known to be accurate in the far tails of the PDF.

3 For a variable reference hypotheses, the J-function is

$$4 \quad J(x, T, H_0(z)) = \frac{p_x(x|H_0(z))}{p_z(z|H_0(z))}. \quad (27)$$

5 Modules using a variable reference are usually designed to

6 position the references hypothesis at the peak of the

7 denominator PDF, which is approximated by the CLT.

8 A special case of the variable reference hypothesis

9 approach is the ML method, when z is an MLE. Whenever the

10 feature is also a ML estimate and the asymptotic results apply

11 (the number of estimated parameters is small and the amount of

12 data is large), the two methods are identical. The variable

13 reference hypothesis method is more general because it does

14 not need to rely on the CLT.

15 One-to-one transformations do not change the information

16 content of the data, but they are important for feature

17 conditioning prior to PDF estimation. Recall from that the PDF

18 projection theorem is a generalization of the change-of-

19 variables theorem for 1:1 transformations. Thus, for 1:1

20 transformations, the J-function reduces to the absolute value

21 of the determinant of the Jacobian matrix (4)

$$22 \quad J(x, T) = |J_T(x)| \quad (28)$$

23 Application of the PDF projection theorem to

24 classification is performed by substituting (18) into (1). In

25 other words, we implement the classical Neyman-Pearson

1 classifier but with the class PDFs factored using the PDF
 2 projection theorem

$$3 \quad j^* = \arg \max_j \frac{p_x(x|H_{0,j}(z_j))}{p_z(z|H_{0,j}(z_j))} \hat{p}_z(z_j|H_j) \text{ at } z_j = T_j(x) \quad (29)$$

4 where we have allowed for class-dependent, variable, reference
 5 hypotheses.

6 FIG. 2 shows an example of a classifier 10 constructed
 7 with the architecture of the current invention. Raw data X
 8 having a plurality of time samples and falling into a
 9 plurality of classes is provided to the classifier 10. Raw
 10 data X is provided to chains 11 of class-specific modules 12.
 11 Each class is associated with a chain 11 of class-specific
 12 modules 12.

13 Each module 12 receives a feature calculation input which
 14 it provides to a feature calculation section 14. The feature
 15 calculations section performs calculations on the feature
 16 calculation input. The feature calculation input can be data
 17 or previously computed features from previous feature
 18 calculation outputs. Upon completing these calculations the
 19 module 12 provides a feature calculation output. Each module
 20 12 also includes a Log J-Function section 16. The Log J-
 21 Function section 16 computes a correction factor that can be
 22 summed at summer 18 with the correction factors provided by
 23 the correction output of Log J-Function sections 16 in
 24 previous modules 12 to allow chaining of modules 12.

25 Modules 12 are joined in chains so that the first module
 26 in the chain receives raw data X at its feature calculation

1 input and zero or a null value at its correction input. Each
2 succeeding module 12 then receives its inputs from the
3 preceding module 12 in the chain 11. Chain 11 can have any
4 number of modules 12. The last module 12 in the chain 11 is
5 joined to a probability density function evaluation section
6 20. The probability density function evaluation section 20
7 receives the feature calculation output from the last module
8 in the chain and converts it into a form for summing at summer
9 22 with the correction output of the last module 12 in the
10 chain 11. The output of summer 22 applies the probability
11 density function for the class associated with the chain 11 to
12 the raw data and produces a value indicating the likelihood
13 that the raw data is a member of the class. A compare module
14 24 is joined to the output of each summer 22. The compare
15 module 24 provides an output that indicates that the raw data
16 X is of the class having features indicated by high values at
17 the outputs of summers 22.

18 Class specific modules 12 have been built for feature
19 transformations including various invertible transformations,
20 spectrograms, arbitrary linear functions of exponential random
21 values, the autocorrelation function (contiguous and non-
22 contiguous), autoregressive parameters, cepstrum, order
23 statistics of independent random values, and sets of quadratic
24 forms. These represent some of the many feature
25 transformations that can be incorporated as modules in a
26 classifier built using the chain rule.

1 FIG. 3 shows an example of a classifier 10' constructed
2 using an alternative embodiment of the architecture of the
3 current invention. This architecture utilizes a J-Function
4 26, instead of a Logarithmic J-Function 18, in each module 12.
5 This J-Function can be multiplied with the previous correction
6 outputs at multiplier 30. The probability function evaluation
7 section 34 can then provide an output which can be multiplied
8 at 32 with the output of the last module. The multiplied
9 output can then be used as the probability density function
10 for the feature.

11 FIG. 4 is another alternate embodiment 10" of a
12 classifier utilizing the architecture taught by the current
13 invention. In this embodiment, a thresholding module 36 is
14 provided for each class between summer 22 and compare module
15 24. Thresholding module 36 does not allow summer 22 to send a
16 value to compare module 24 if the value does not exceed a
17 threshold value. This threshold value can be set as one value
18 for all of the chains or set independently for each chain.
19 The threshold value can be calculated based on the level of
20 background noise in the raw input data. Use of thresholding
21 modules 36 allows weak samples to be ignored rather than
22 forcing them into a poorly fitting class. While thresholding
23 is shown applied to the log J-function embodiment, it can also
24 be applied to the J-function embodiment of the invention.

25 The J-function and the feature PDF provide a
26 factorization of the raw data PDF into trained and untrained
27 components. The ability of the J-function to provide a "peak"

1 at the "correct" feature set gives the classifier a measure of
 2 classification performance without needing to train. In fact,
 3 it is not uncommon that the J-function dominates, eliminating
 4 the need to train at all. This we call the *feature selectivity*
 5 *effect*. For a fixed amount of raw data, as the dimension of
 6 the feature set decreases, indicating a larger rate of data
 7 compression, the effect of the J-function compared with the
 8 effect of the feature PDF increases. An example where the J-
 9 function dominates is a bank of matched filter for known
 10 signals in noise. If we regard the matched filters as feature
 11 extractors and the matched filter outputs as scalar features,
 12 it may be shown that this method is identical to comparing
 13 only the J-functions. Let $z_j = |\mathbf{w}'_j \mathbf{x}|^2$, where \mathbf{w}_j is a normalized
 14 signal template such that $\mathbf{w}'_j \mathbf{w}_j = 1$. Then, under the white
 15 (independent) Gaussian noise (WGN) assumption, z_j is
 16 distributed $\chi^2(1)$. It is straightforward to show that the J-
 17 function is a monotonically increasing function of z_j . Signal
 18 waveforms can be reliably classified using only the J-function
 19 and ignoring the PDF of under each hypothesis. The curse of
 20 dimensionality can be avoided if the dimension of z_j is small
 21 for each j . This possibility exists, even in complex problems,
 22 because z_j is required only to have information sufficient to
 23 separate class H_j from a specially chosen reference hypothesis
 24 $H_{0,j}$.

1 This invention has been disclosed in terms of certain
2 embodiments. It will be apparent that many modifications can
3 be made to the disclosed apparatus without departing from the
4 invention. Therefore, it is the intent of the appended claims
5 to cover all such variations and modifications as come within
6 the true spirit and scope of this invention.